جامعة طرابلس – كلية تقنية المعلومات

تحليل وتصميم النظم
Systems Analysis & Design

أستاذة المقرر
أ.صبرية عبد القادر المصراتي

Use Case Diagram

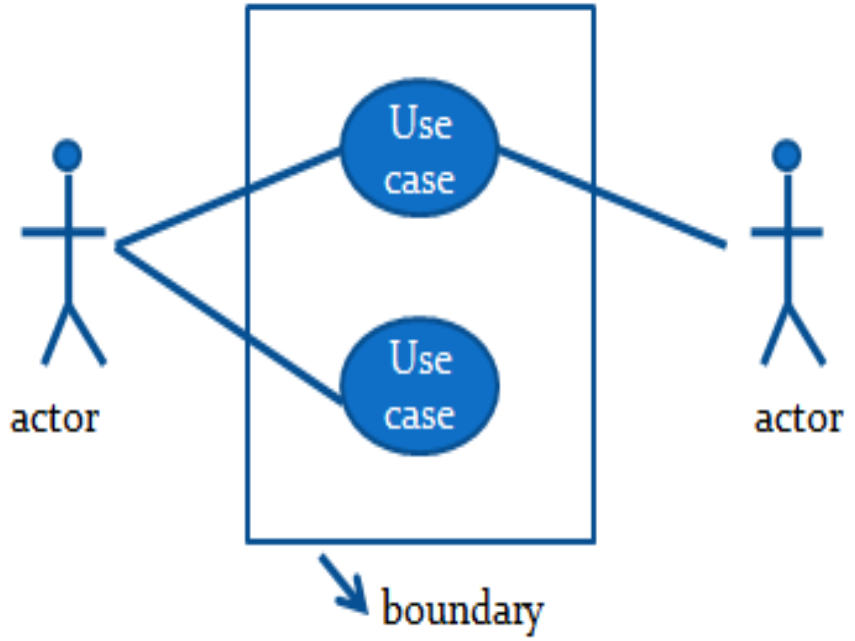
مخطط حالة الاستخدام هو أداة تحليل كائنيه تستخدم لغرض تحديد المتطلبات الوظيفية للنظام ونطاق المشروع ويمكن استخدامه في النظم الهيكلية والكائنية على حد سواء.

وهي يمكن أن تستخدم تقريبا في كل المشاريع كوسيلة اتصال بين المستخدم ومحلل النظم عند تعريف وتحديد المتطلبات الوظيفية.

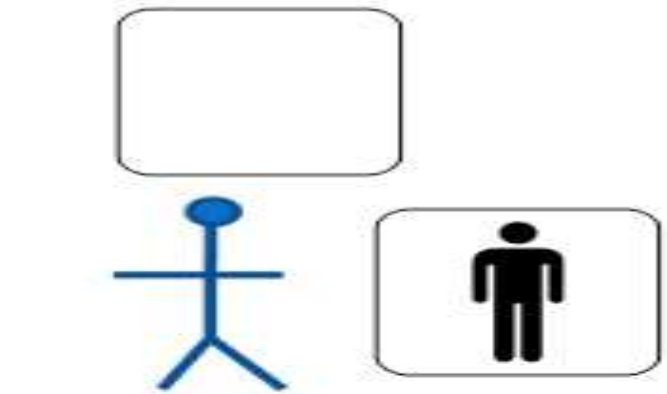
UCD هي إحدى أدوات UML (Unified Modelling Language) في التحليل ,وتعنى UML لغة النمذجة الموحدة. وهى عبارة عن مخططات ونصوص لغرض إعداد البرمجيات الكائنية.

Use Case Diagram

يتكون UCD من حالات استخدام (وظائف) وممثلين (مستخدم أو نظم خارجية) في تفاعل مع بعضهم البعض لإنجاز معاملات النظام كما بالشكل التالي:



Symbols of Use Case Diagram



حدود المنظومة
Scope

المستخدم
User

وظيفة
Function

علاقة اجبارية
include

علاقة اختيارية
Extind

Symbols of Use Case Diagram

1. ممثل (Actor) :

الممثل أما يكون مستخدم أو نظام خارجي يتفاعل مع النظام المقترح.

مثال :

الممثل يمكن أن يلعب أحد الادوار التالية:

- بائع (salesman)
- مكتبي (librarian)
- زبون (customer)
- نظام الدفع (billing system)

<u>لاحظ أن :</u>

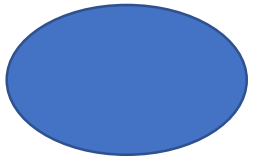
- الممثل الواحد يمكن أن يمثل عدة مستخدمين أو أنظمة خارجية .
- المستخدم الواحد أو النظام الخارجي الواحد يمكن أن يلعب أدوار متعددة.
- من المهم أن نفرق بين المستخدم والممثل : المستخدم هو أنسان يقوم باستخدام النظام أما الممثل فيمكن أن يكون مستخدم أو نظاما خارجيا يتصل بالنظام المقترح.

Symbols of Use Case Diagram

2. حالة استخدام (Use Case)

هي عبارة عن إجراء يتفاعل الممثلون عن طريقه مع النظام وهو بلغة تحليل النظم الهيكلية (عبارة عن معاملة أو وظيفة).

الشكل التالي يمثل حالة استخدام يكون بيضاويا على النحو التالي :



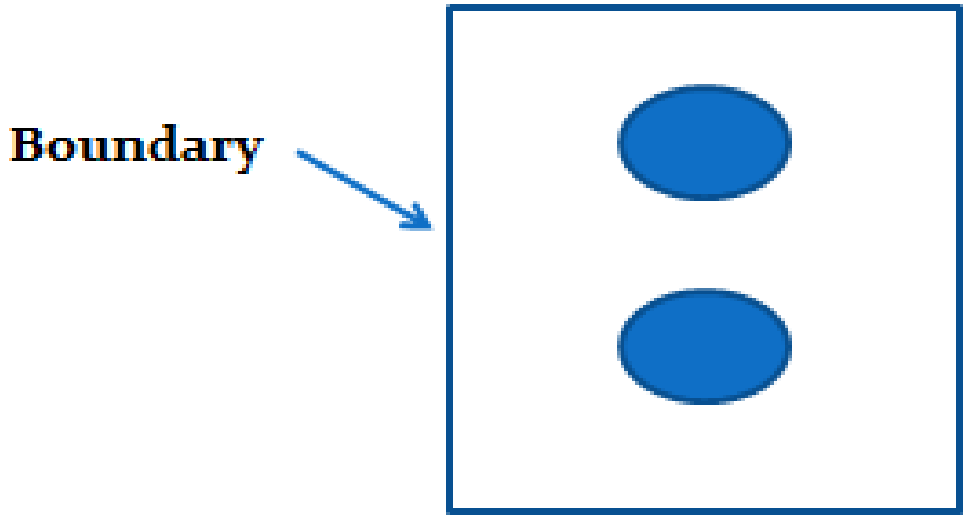
الاجراء الذى يتم تنفيذه في هذه الحالة يكتب داخل الشكل البيضوي كما يلى :

لكل ممثل يوجد على الاقل حالة استخدام واحدة.

Symbols of Use Case Diagram

3. الحدود (Boundary)

الحدود تبين نطاق النظام ونمثلها في المخطط بشكل مستطيل يحيط بالنظام ويفصله عن بيئته الخارجية .



Levels Use Case Diagram & Relationships

1. **المستوى 0 البيئي أو التفاعلي :**
2. **المستوى 1 التفصيلي :** الذي ينقسم إلى 3 أنواع من العلاقات وهي :

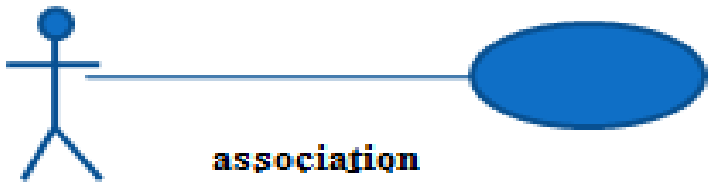
- علاقة ممثل وحالة استخدام تسمى علاقة ربط (association).
- علاقة حالة استخدام وأخرى تسمى علاقة امتداد أو شمول (extend or include).
- علاقة ممثل بأخر أو حالة استخدام مع أخرى تسمى علاقة تعميم (generalization).

Relationships

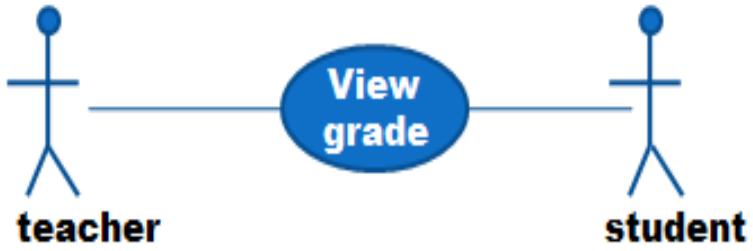
4. العلاقات (Relationships)

1. علاقة ربط (association)

- يتم تمثيل العلاقة بخط يربط بين الممثل وحالة (أو حالات)الاستخدام وذلك لتوضيح أن الممثل هو الذى يقوم بحالة الاستخدام.



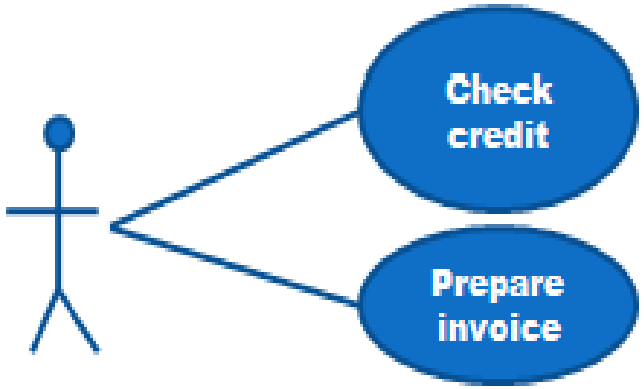
- حالة الاستخدام التى يقوم بها ممثلون عديدون يتم توضيحها كما يلي :



Relationships

في الشكل التالي الممثل (البائع) يقوم بحالات استخدام متعددة:

- افحص الرصيد (check credit)
- جهز الفاتورة (prepare invoice)



Relationships

2. علاقة امتداد / شمول Extend /Include Relationship

يمكن أن توجد علاقة بين حالة استخدام وأخرى.

وهناك نوعان من العلاقة بين حالات الاستخدام وهما:

1. علاقة الشمول **(Include Relationship)**

2. علاقة الامتداد **(Extend Relationship)**

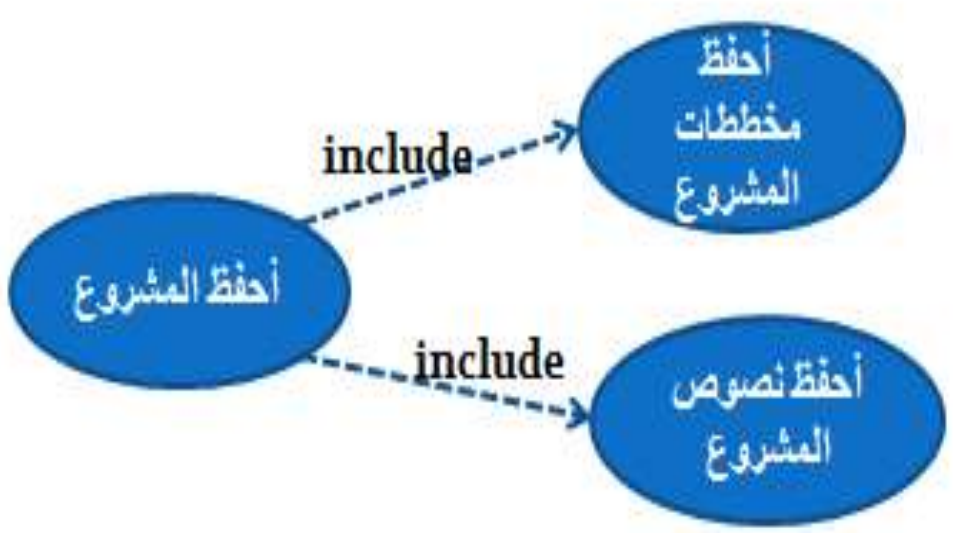
➢ **علاقة الشمول (Include Relationship)**

تستخدم لتبين أن حالة استخدام (تسمى إما الأساسية أو الأب) تشمل وظائف موجودة في حالة (أو حالات) استخدام أخرى (تسمى الابناء).

Include Relationship

مثال 1:

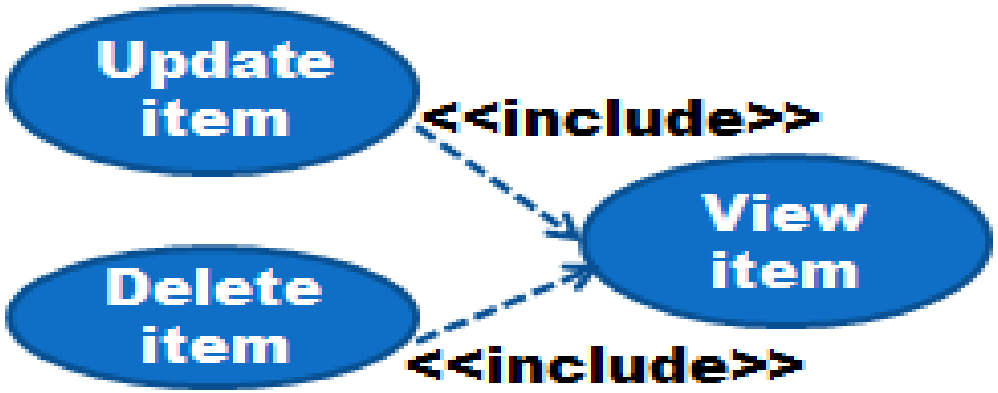
في الشكل التالي نجد أن حالة الاستخدام (احفظ المشروع) تعتبر أساسا (أب) بينما تعتبر حالة الاستخدام (احفظ مخططات المشروع) وحالة الاستخدام (احفظ نصوص المشروع) أبناء لحالة الاستخدام الاساسية.



Include Relationship

مثال 2:

في المخطط التالي لدينا حالة استخدام (اعرض الصنف) مستخدمة من قبل حالتي استخدام وهما (عدل الصنف) و (احذف الصنف).



Extend Relationship

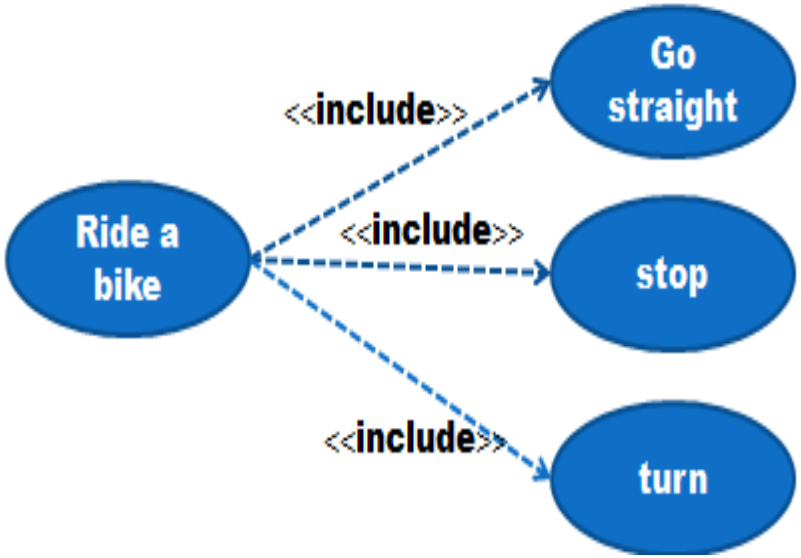
➢ **علاقة الامتداد (Extend Relationship)**

علاقة الامتداد بين حالة استخدام وأخرى تحدث عندما يكون لدينا أحد الأوضاع التالية:

- حالة استخدام رئيسية لديها حالات استخدام بديلة (خيارات).
- حالة استخدام إضافية يمكن إضافتها لحالة الاستخدام الرئيسية.

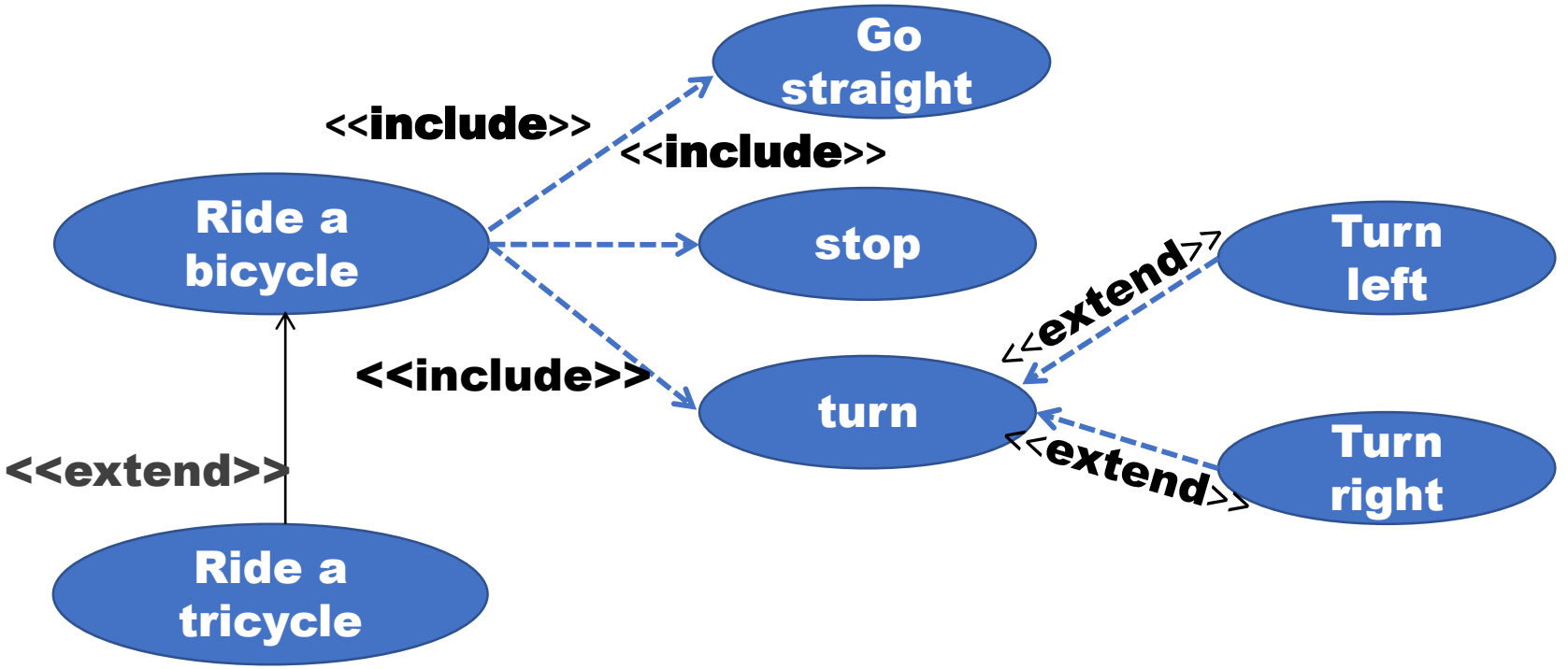
مثال 3:

حالة الاستخدام التالية (اركب دراجة):



Extend Relationship

يمكن شرحها باستخدام علاقة امتداد كما يلى :

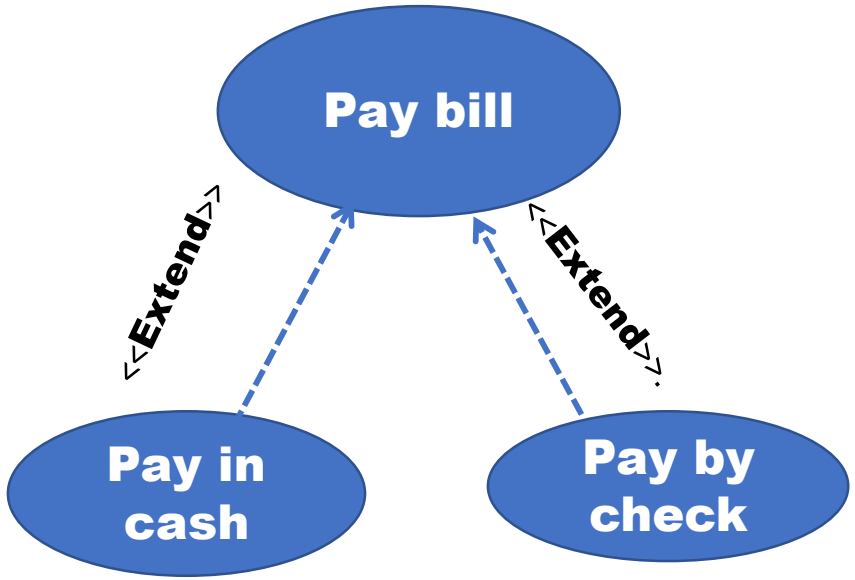


Extend Relationship

مثال 4:

حالة الاستخدام التالية (ادفع الفاتورة) في نظام مبيعات يمكن أن يكون لها حالتي استخدام بديلة كما يلي :

- الدفع نقدا **(pay in cash use case)**
- الدفع بصك **(pay by check use case)**



Use Case Description

➢**وصف مسألة نظام دراسة مبسط**

في كلية التقنية الالكترونية يقوم الطلاب بالتسجيل في المقررات ,ويقوم المسجل بعملية التسجيل أما أعضاء هيئة التدريس فيرصدون درجات الطلاب . ومن إحدى اختصاصات المسجل أن يعطى الطالب كشفا بجميع درجاته إذا طلب منه ذلك .

Actors في هذا النظام هم:

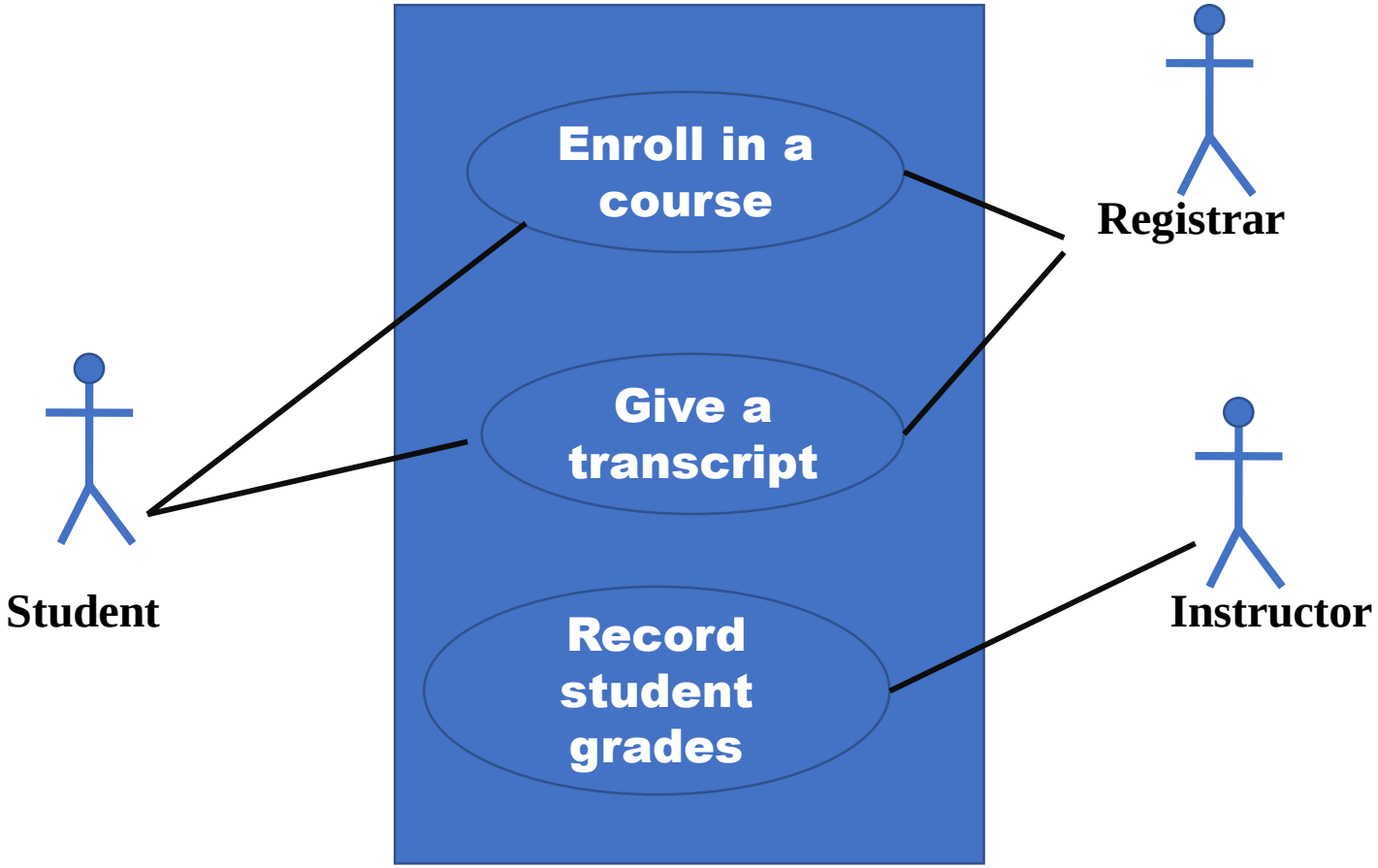
- الطالب (Student)
- المسجل (Registrar)
- المعلم (Instructor)

The use cases حالات الاستخدام:

- سجل في مقرر (Unroll in a course).
- ارصد درجات طالب (Record student grades).
- امنح كشف درجات (Give a transcript).

ارسم مخطط حالة الاستخدام لهذا النظام مستخدما المعلومات في هذا السيناريو :

مخطط حالة الاستخدام للنظام تكون كما يلي:

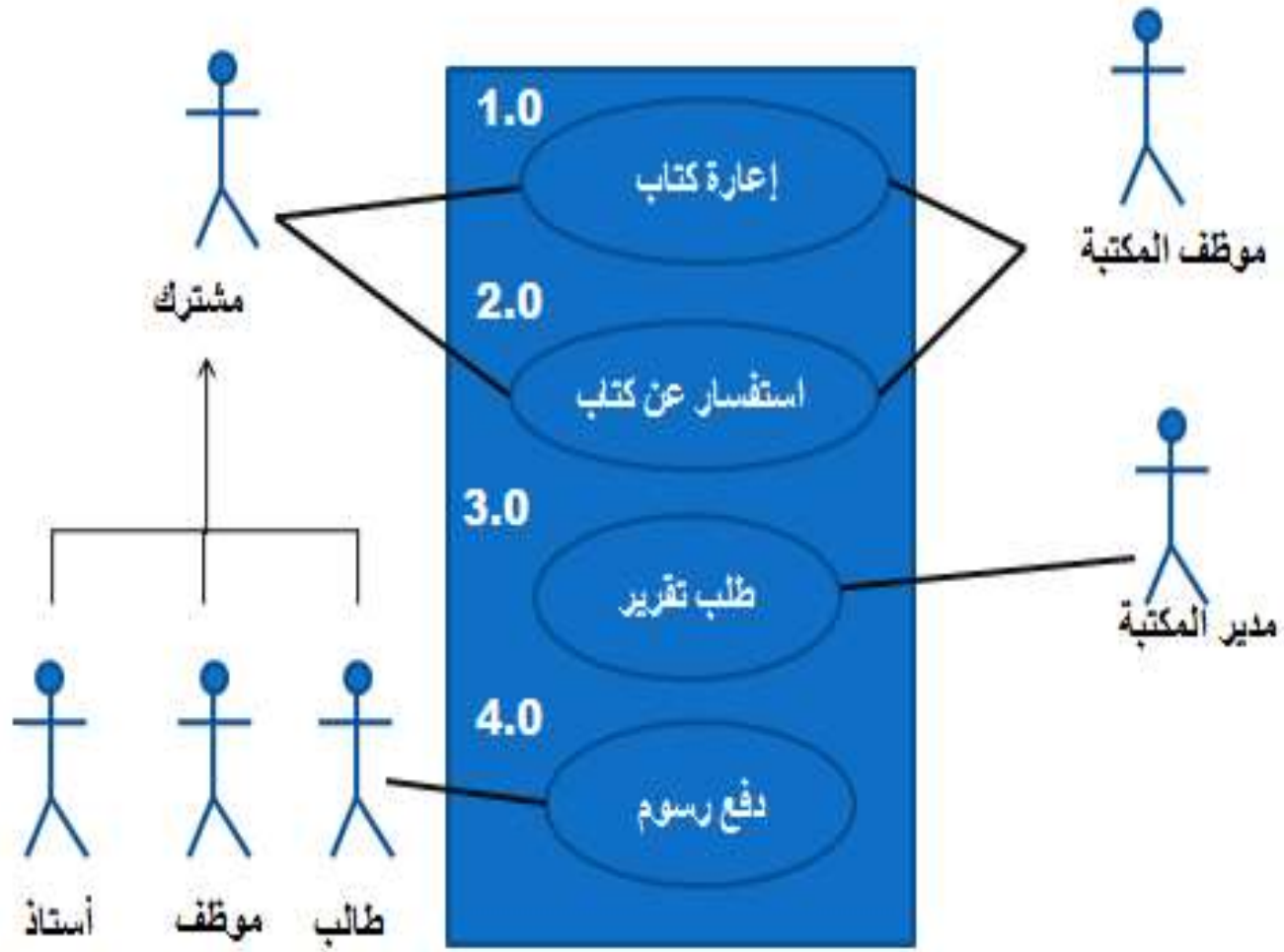


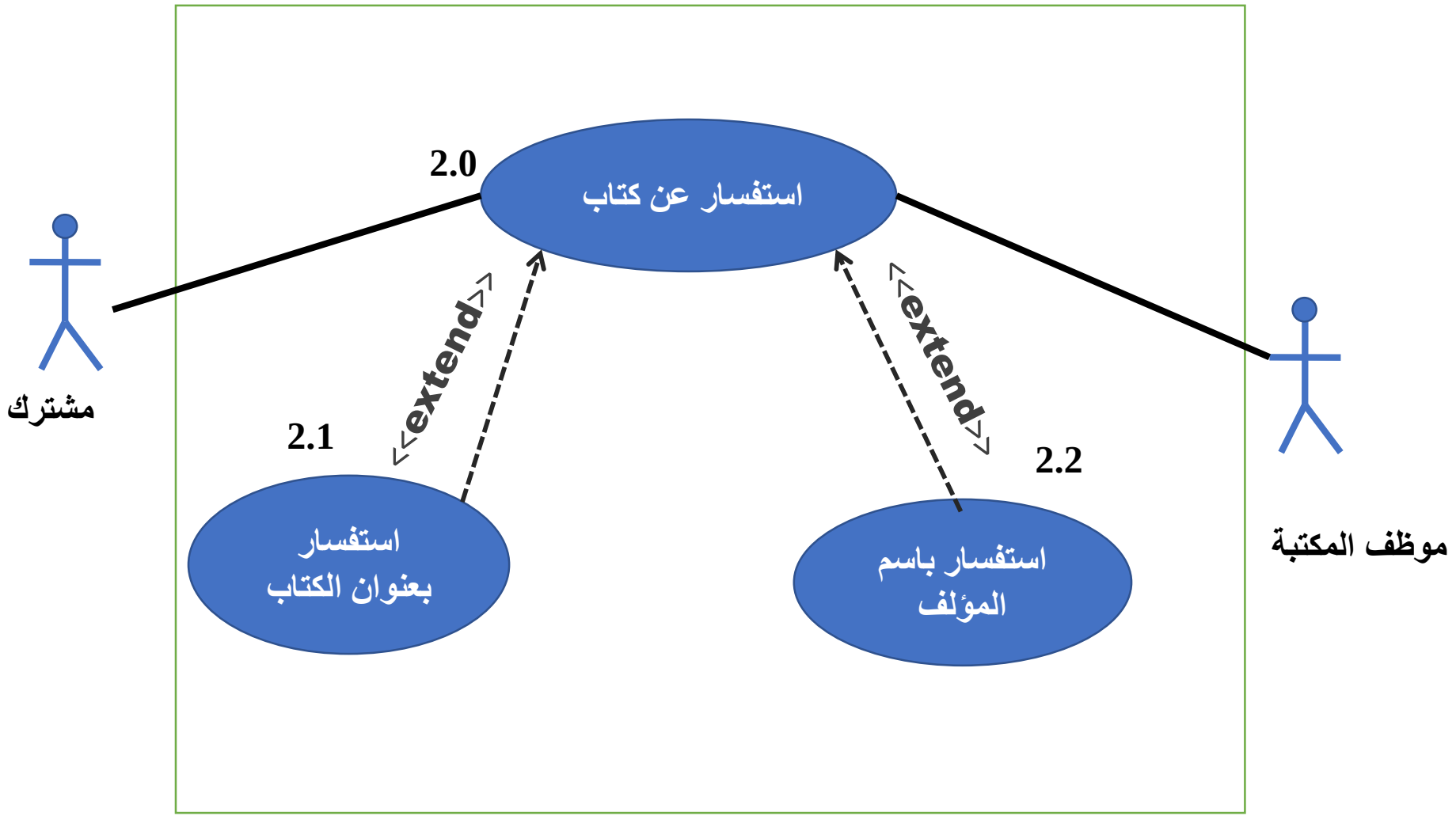
Use Case Description

➢**وصف مسألة نظام المكتبة**

في مكتبة الكلية يقوم الطلاب والموظفين والمدرسين باستعارة الكتب , ويقوم موظف المكتبة بعملية الإعارة. ويتحصل مدير المكتبة على تقارير يومية عن الكتب المستعارة والمرجعة. ويقوم المشتركين بالاستفسار عن الكتب أما بواسطة عناوين الكتب أو المؤلفين ,ويقوم موظف المكتبة بعملية الاستفسار ويقوم الطالب بدفع رسوم نظير الاستعارة.

- حدد الممثلين (actors) في النظام.
- حدد حالات استخدام النظام.
- ارسم مخطط حالات الاستخدام (UCD).


1.0
إعارة كتاب
2.0
استفسار عن كتاب
3.0
طلب تقرير
4.0
دفع رسوم
مشترك
موظف المكتبة
مدير المكتبة
أستاذ
موظف
طالب


2.0
استفسار عن كتاب
مشترك
<<extend>>
<<extend>>
2.1
2.2
استفسار
بعنوان الكتاب
استفسار باسم
المؤلف
موظف المكتبة

Decision Tree

Decision Tree

تعتبر **شجرة القرار** تمثيل تخطيطي لعملية اتخاد القرار وتعتبر شجرة القرار من الوسائل التي يستخدمها محلل النظم في تحليل وفهم المشكلة المطروحة وتستخدم عندما تكون المشكلة مركبة وذات قيمة احتمالية تحتوى على عدد كبير من البدائل.

ملاحظة :

1. تقرا من شمال الجذر إلى اليمين.
2. كل فرع يوضح شرط يمكن حدوثه.
3. تركيبة الشروط توصلنا إلى أفعال في الجهة اليمنى.
4. يمكن استخدامها كبديل لجدول القرار.

Decision Tree

استخدامات شجرة القرار

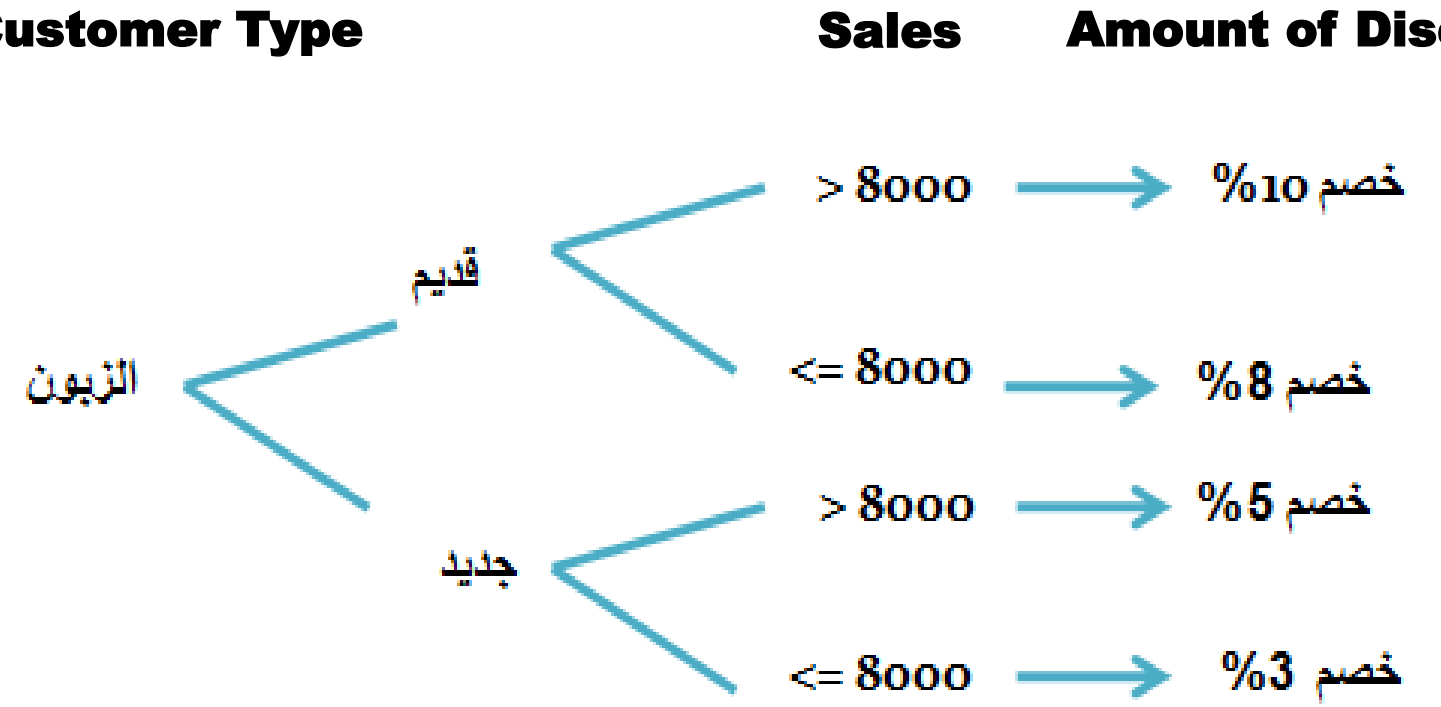
1. تتعامل شجرة القرار مع السياسات ذات التفرع المتعدد مثل حساب تخفيضات المبيعات ومكافئات الانتاج وحساب الاجور.
2. تستخدم عندما يكون عدد الافعال صغيرا ويمكن توضيح كل الاحتمالات.
3. من الافضل استخدامها للتحقق من المنطق وعندما تكون القرارات غير معقدة.

Decision Tree

مثال :

يحصل **الزبون القديم** على خصم 10% إذا كانت قيمة المشتريات أكبر من 8000 والا 8%.

يحصل **الزبون الجديد** على خصم 5% إذا كانت قيمة مشترياته أكبر من 8000 والا 3%.



Decision Table

جدول القرار هو عبارة عن أداة تحليل وتصميم ,تبين المنطق الذى يربط بين الحالات والافعال ,وهو عبارة عن بنية مترابطة من الصفوف والاعمدة تمثل الصفوف كل الشروط (Conditions) والافعال (Actions) وتمثل الاعمدة قواعد القرار(Decision Rules)المختلفة.

استخدامات جدول القرار

1. تستخدم في تطبيقات الادارة والتحكم.

2. تستخدم عندما تكون الافعال معتمدة على مجموعة كبيرة من الشروط تؤخذ علي مجموعات جزئية معتمدة.

3. يمكن استعمالها في التطبيقات التي تتعلق بالتصنيع ,التأمين ,المالية, البحث والتطوير.

Components of Decision Table

الشروط (Condition):

عبارة عن حقائق أو أحداث تقرر الأفعال التي يجب تنفيذها.

الأفعال (Action):

هي العمليات أو المعالجات التي تنفذ تحت شروط معينة.

	قواعد القرار Decision Rules				
شروط Conditions					
أفعال Actions					

قاعدة القرار (Decision Rule):

وهي العلاقات بين الشروط والأفعال.

Y = الحالة تحقق الشرط.

N = الحالة لا تحقق الشرط.

X = الشرط يحقق الحدث أو الفعل.

مثال (1) :

صمم جدول قرار لوصف المسألة التالية :

يتم توزيع طلبة الكلية الذين اجتازوا الفصل الثاني بنجاح على تخصصات الكلية الثلاث (الحاسب/التحكم/الاتصالات) حسب رغبتهم بناء على القواعد التالية :

1. ينسب الطالب إلى قسم الحاسب الآلي إذا كان معدله أكبر من أو يساوي 70 % و درجته النهائية في مقرر رقمية 1 أكبر من 65%.

2. ينسب الطالب إلى قسم الاتصالات إذا كان معدله أكبر من أو يساوي 70 % و درجته النهائية في مقرر كهربية 1 أكبر من 65%.

3. في الحالات الأخرى ينسب الطالب إلى قسم التحكم الآلي.

R8	R7	R6	R5	R4	R3	R2	Rule1	
N	N	N	N	Y	Y	Y	Y	المعدل >= 70
N	N	Y	Y	N	N	Y	Y	درجة الرقمية >= 65
N	Y	N	Y	N	Y	N	Y	درجة الكهربية >= 65
-	-	-	-	-	-	X	X	ينسب لقسم الحاسب
-	-	-	-	-	X	-	X	ينسب لقسم الاتصالات
X	X	X	X	X	-	-	-	ينسب لقسم التحكم

Decision Table

مثال (2) : المطلوب إعداد **جدول القرار** لحساب مرتب موظف حيث المرتب يساوى مجموع المرتب الأساسي ومكافأة العمل الإضافي وهى تحسب فقط إذا كانت ساعات العمل أكبر من 40 ساعة غير ذلك يحسب المرتب الاساسي فقط.

Condition	Decision Rule		
	1	2	3
Hours-worked <40	Y	N	N
Hours-worked =40	N	Y	N
Hours-worked >40	N	N	Y
Actions			
Compute basic pay	X	X	X
Compute over-time	-	-	X

الجدول أعلاه يمثل استعمال جدول القرار لتوضيح سياسة حساب راتب موظف.

Decision Table

- لاحظ في الجدول السابق استعمال الرموز التالية :

- **Y :** Yes
- **N :** No
- **X :** Applicable
- **_ :** Not Applicable

- نلاحظ أيضا أن عدد قواعد القرار يعتمد على كل من عدد الشروط وعدد الأفعال.

Decision Table

مثال (3) : تحديد نسبة الخصم على أساس ثلاثة عوامل :-

1. القيمة الإجمالية للطلبية يكون الخصم 3% للطلبيات التي تزيد قيمتها عن (4000) دينار.
2. المسافة بين المخزن ومكان التسليم لا تزيد عن 50 كم يمنح العميل أو الزبون خصم 2% إذا لم يتمتع بخصم الكمية والا 1%.
3. العملاء الذين تزيد قيمة مشترياتهم عن 100 الف دينار خلال السنة الماضية تحصل على خصم إضافيا 2%.
4. مجموع الخصومات 6%

ملاحظة : عدد الأعمدة يحسب 2^n حيث n عدد الشروط.
فى هذا المثال $2^3 = 8$,أي 8 أعمدة.

Decision Table

جدول القرار لسياسة خصم مشتريات لشركة س

الشروط	قواعد القرار							
القيمة < 4000	Y	Y	Y	Y	N	N	N	N
المسافة => 50 كم	Y	Y	N	N	Y	Y	N	N
مشتريات السنة الماضية > 100الف	Y	N	Y	N	Y	N	Y	N
الافعال								
%0								X
%1								
%2						X	X	
%3				X				
%4		X			X			
%5			X					
%6	X							

Decision Table

مثال (4): جدول القرار لتقدير الطالب حسب مقياس 4

جدول القرار الخاص بالتقديرات					
الشرط	1	2	3	4	5
الدرجة من 3.4 فأكثر	Y	N	N	N	N
الدرجة من 3.0 إلى أقل من 3.4	N	Y	N	N	N
الدرجة من 2.6 إلى أقل من 3.0	N	N	Y	N	N
الدرجة من 2.0 إلى أقل من 2.6	N	N	N	Y	N
إلى اقل من 2.0	N	N	N	N	Y
الحدث					
ممتاز	X	.	.	.	.
جيد جدا	.	X	.	.	.
جيد	.	.	X	.	.
مقبول	.	.	.	X	.

Decision Table

➢مزايا جداول القرار

1. وسيلة لوصف سياسة عمل النظام.
2. تغطية كافة الحالات المتوقعة.
3. تشرح العمليات بطريقة شروط وتحدد أفعالها.
4. تبنى أوجه التضارب والنقص فى القواعد.

➢ عيوب جداول القرار

1. غير عملية فى حالة عدد كبير من الشروط.
2. غير مناسبة فى حالة التشابه بين الشروط وطرق التنفيذ.
3. صعوبة تكوينها خاصة لمن يستعملها لأول مرة.

Structured English

أداة تحليل نصية تستخدم جزء محدود من اللغة الانجليزية لتوضيح الخطوات المراد أداؤها لوظائف (عمليات) نظام معين.

- **استعمالات الانجليزية المركبة :**

تستعمل الانجليزية المركبة في المسائل التي تدمج بين سلسلة من الاعمال والحسابات مع اختبارات وتكرارات.

Rules in Writing Structured English

1.توضيح منطق سير العمليات خطوة خطوة ,وذلك بصيغة جمل بسيطة بصيغة أفعال الأمر.

2. يجب تحليل الجمل المركبة إلى عناصرها البسيطة.

3. استعمل مصطلحات من قاموس البيانات (Data Dictionary) ومخطط انسياب البيانات (DFD).

4. يجب اختيار جمل واضحة وقياسية غير قابلة للتأويل ,وأن تكون مألوفة ومناسبة.

5.استعمل حروفا كبيرة في الكلمات المفتاحية , CASE, IF ,THEN , ELSE REPEAT

Structured English

➢مصطلحات الانجليزية المركبة :

1. القائمة التالية تحتوى على أفعال مناسبة تستعمل في الانجليزية المركبة

Compute	Sort
Merge	Print
Determine	Display
Verify	Format
Check	Retrieve
Update	Compare
Prepare	Edit
Enter	Move
Generate	Read
Add	

Structured English

- مصطلحات الانجليزية المركبة :

2. الكلمات المعرفة في قاموس البيانات مثل :

Item-price

Student-name

Number-of-Students

Data-of-last-transaction

Structured English

➢مصطلحات الانجليزية المركبة :

3. الكلمات المحجوزة المستعملة في البرمجة الهيكلية مثل :

IF _THEN ELSE

CASE

DO - WHILE

REPEAT - UNTIL

Structured English

❖ **أنماط أو مصطلحات اللغة المركبة (الانجليزية المركبة)**

1. النمط المتسلسل (هيكل العبارات المتتابعة)

يعبر عن اجراءات متتابعة تكون على صورة افعال امر و مفعول به مثل **احسب الضريبة ,**

حدد ضريبة الدخل.

Structured English

2. نمط القرارات : تتضمن شروط تحديد حركة سير الاجراءات وهناك نوعين :

أ. **ذو الاتجاهين** وذلك للشروط ذو الاحتمالين ويعبر عنها بالصور

IF (condition)

THEN(blocke1)

ELSE(blocke2)

ب. **متعدد الاتجاهات** : عندما يكون هناك عدد من البدائل ويعبر عنه بصورة

SWITCH (variable)

CASE1(value 1) do block1

CASE2(value 2) do block2

CASE3(value 3)

Structured English

❖ **Iteration : النمط التكراري**

سلسلة أحداث تكرار بناء على شروط مبنية وتكون على الصور الآتية:-

DO WHILE(condition)
DO block

REPEAT
DO block
UNTIL (condition)

FOR, for value

Structured English

مثال (1)

المثال التالي يبين سياسة حساب التخفيض باستخدام الانجليزية المركبة :

IF customer type= ‘old customer’ THEN

لو نوع الزبون =”زبون قديم“ فان

IF amount of sales is more than 10,000

لو مقدار المبيعات أكبر من 10,000

THEN discount = 10% فإن التخفيض

ELSE discount = 8% والا فإن التخفيض

ELSE **و الأ**

IF amount of sales is more than 10000

لو مقدار المبيعات أكبر من **10,000**

THEN discount =5% فأن التخفيض

ELSE discount =3% والا فالتخفيض

مقارنة لدواعي استخدام أدوات التحليل لوصف سياسة عمل المنظومة

الانجليزية المهيكلة	جدول القرار	شجرة القرار
تستعمل للمسائل التي تحتوى على الحلقات والقرارات والسلسة من الأفعال	تستعمل للمسائل ذات المنطق المركب من عدة شروط	تستعمل للقرارات البسيطة

من هذا الجدول نستخلص أنه إذا كانت المسألة التي لدينا بسيطة يمكننا أن نستعمل شجرة القرار , أما إذا كانت المسألة معقدة فنستعمل إما جدول القرار أو الانجليزية المركبة.

ملاحظة : لاحظ أن الانجليزية المركبة أو العربية المركبة يمكن فهمها من قبل المحلل والزبون على حد سواء ولهذا يمكن أن تكتب بأي لغة مثل العربية.

Phases and Activities of System Development Life Cycle (2)

(Activities)	(Phase)
• التشفير (Coding) • اكتشاف الاخطاء (Debugging) • اختبار الوحدة (Unit Test)	Implementation
• اختبار التكامل (Integration Test) • اختبار النظام (System Test) • اختبار القبول (Acceptance Test)	Testing
• التحسين (Enhancement) • التكيف (Adaptation) • التصحيح (Correction) • إعادة الهندسة (Re-engineering)	Maintenance

Implementation Phase

- الهدف الرئيسي لمرحلة التنفيذ هي إنتاج شفرة مصدر (Source code) التي عند تنفيذها نحصل على منظومة تعمل بكفاءة (بدون أخطاء).
- وضوح شفرة المصدر(Source code) يسهل اكتشاف الاخطاء والاختبار والتعديل.

مرحلة التنفيذ تتعلق بالنشاطات التالية :

- ترجمة مواصفات التصميم لكل جزء برمجي (Module) إلي شفرة مصدر (Source code).
- ترجمة البرنامج والبحث وتصحيح الاخطاء.
- إجراء اختبار للوحدات (Unit Test) أي للأجزاء البرمجية (Module Test).

ملاحظة : يعتمد عدد المبرمجين في هذه المرحلة على حجم المشروع وهيكلية الفريق الذي تم تشكيله عند تخطيط المشروع.

Implementation Phase

عند البرمجة يجب تحقيق الاتي : -

- استعمل لغة البرمجة التي تناسب التطبيق المطلوب.
- اجعل قابلية قراءة المخرجات (Output) أفضل ما يمكن باستعمال الواجهة الرسومية (GUI).
- اجعل قابلية قراءة شفرة المصدر(Source code) أفضل ما يمكن باستعمال أسماء ذات معنى وجمل تعليقيه (Comment Statements).
- اجعل وقت إعداد المنظومة أقل ما يمكن باستخدام لغة برمجة ذات كفاءة عالية.
- اجعل عدد جمل شفرة المصدر(Source code) أقل ما يمكن.
- أجعل الذاكرة (Memory) المطلوبة أقل ما يمكن.

Implementation Phase

يجب أن يحقق الجزء البرمجي ما يلي:-

- أن يكون له مدخل واحد ومخرج واحد.
- لا يوجد به جمل GOTO إلا للضرورة.
- أن يحتوي على أقل من 30 جملة (أي حوالي صفحة).
- أن يستعمل التنسيق والاسطر الخالية لجعل الجزء البرمجي هيكلياً وواضحاً.
- ألا يتداخل بعمق كبير (مثلا لا يزيد عن 5 جمل شرطية (if-then-else) أو حلقة (while loops) متداخلة).
- أن يحتوي على توثيق (Documentation) كافي.

ملاحظة : في هذه المرحلة يجب أن يتم إعداد كتيب المشغل (User manual) لمعرفة كيفية تشغيل المنظومة ويتم استعماله في المراحل اللاحقة.

Testing Phase

إن الهدف الرئيسي من الاختبار هو تحديد إمكانية الحصول على النتائج المطلوبة عند تشغيل المنظومة.

أن اختبار المنظومة الجديدة يجب أن يحقق:-

1. التخلص من الاخطاء اللغوية (Syntax errors).
2. التخلص من الاخطاء المنطقية (Logic errors).
3. التخلص من الاخطاء التنفيذية (Run time errors).
4. تحديد أخطاء مدخلات المستخدم (User input errors).

Testing Phase

أن اختبار المنظومة الجديدة يجب أن يحقق:-

5. تقييم أمن المنظومة.

6. تقييم سرعة أداء المنظومة.

7. اكتشاف أي وظيفة مفقودة.

8. تحقيق متطلبات المستخدم (User Requirements).

9. تقييم توثيق المنظومة.

Testing Phase

لاختبار منظومة جديدة عادة يتم استعمال عينة من البيانات ونقارن المخرجات (النتيجة) مع النتيجة المتحصل عليها من نفس العينة يدويا.

يجب أن تحتوي هذه العينة على جميع الحالات المحتملة للتأكد من عدم وجود أخطاء منطقية.

كل جزء برمجي (Module) (برنامج فرعي - Procedure) من المنظومة يتم اختباره منفصلا قبل اختبار المنظومة ككل.

الاختبار يشمل :-

- اختبار التكامل (Integration Test)
- اختبار النظام (System Test)
- اختبار المستخدم (User Test)

Testing Phase

➢ **اختبار التكامل (Integration Testing)**

يتم ربط واختبار الاجزاء البرمجية التي تم اختبارها منفردة مسبقا كنظام متكامل (وحدة واحدة) للتأكد من أن المتطلبات كما عرفها المستخدم قد تم تحقيقها.

يوجد استراتيجيتان شائعتان لاختبار التكامل هما :

- استراتيجية من تحت إلي فوق (Bottom-up strategy)
- استراتيجية من فوق إلي أسفل (Top-down strategy)

Testing Phase

- **اختبار النظام (System Testing)**

يتم التحقق من جميع المتطلبات باستخدام عتاد (Real hardware) وبيانات حقيقية.

- **اختبار القبول (Acceptance Testing)**

يتم تنفيذ الاختبار من قبل الزبون أو المستخدم لا ثبات أن المتطلبات المدونة في وثيقة المتطلبات قد تم أنجازها.

بعد الانتهاء من الاختبار تصبح المنظومة أو المنتج البرمجي جاهز للعمل.

Maintenance Phase

الصيانة هي عملية جعل المنظومة تعمل بطريقة صحيحة في مواجهة العوامل التالية التي قد تؤثر على عملها:-

- التغيرات بسبب أخطاء تحدث بعد عملية تسليم المنظومة.
- استخدام تقنيات جديدة.
- منع أي مشاكل قد تحدث في المنظومة بسبب رداءة في التصميم (إعادة هندسة المنظومة (reengineering)
- تحسين قدرات المنظومة بإضافة وظائف جديدة.

تبدأ عملية الصيانة عندما يتم تسليم المنظومة وتركيبها (تحميلها) في موقع الزبون والبدء في تشغيلها.

Maintenance Phase

النشاطات التي يتم انجازها في مرحلة الصيانة :-

- **تصحيح الاخطاء (Correction)**

يتم في هذا النشاط تصحيح الاخطاء التي تظهر أثناء عملية تشغيل المنظومة والتي لم يتم اكتشافها في مرحلة الاختبار.

- **التكييف (Adaptation)**

نظرا للتغيرات السريعة في التقنية (Software & Hardware) قد تحتاج المنظومة التكيف مع بيئة جديدة مثل منظومة تشغيل جديدة أو معدات حاسوب جديدة.

Maintenance Phase

النشاطات التي يتم انجازها في مرحلة الصيانة :-

- **التحسين (Enhancement)**

يشمل إما تطوير إمكانيات المنظومة أو توفير وظائف أكثر للمنظومة عند اكتشاف متطلبات جديدة.

- **إعادة الهندسة (Re-engineering)**

يتم إعادة تصميم النظام وبرمجته إذا لزم الامر لمنع أي مشاكل متوقعة من التصميم السيئ.